What is claimed is:

1. A method for obtaining at least one device service on a self-service transaction terminal using a service provider framework, comprising:

receiving data by a self-service transaction terminal application indicative of a need for obtaining a transaction terminal device service;

issuing a request by the transaction terminal application to an XFS manager to get the transaction terminal device service;

translating the request by the XFS manager for processing by a service provider; and

obtaining the transaction terminal device service by the service provider.

2. The method of claim 1, wherein receiving the data by the transaction terminal application further comprises receiving the data indicative of the need for obtaining the transaction terminal device service in connection with a device selected from a group of transaction terminal devices consisting of a depository, a printer, a card reader, a safe door, a cash dispenser, and a touchscreen.

3. The method of claim 1, wherein issuing the request to the XFS manager further comprises making a sub-routine call by the transaction terminal application to the XFS manager to get the transaction terminal device service from a service provider.

4. The method of claim 1, wherein translating the request for processing by the service provider further comprises translating a sub-routine call by the XFS manager as an entry point into the service provider for processing by the service provider.

5. The method of claim 4, wherein translating the sub-routine call as an entry point into the service provider further comprises instantiating a request object associated with the request.

6. The method of claim 1, wherein translating the request for processing by the service provider further comprises implementing the service provider.

7. The method of claim 6, wherein implementing the service provider further comprises instantiating an instance of a service provider framework XFS service provider base class and at least one instance of a service provider framework request object required to process the request.

8. The method of claim 7, wherein instantiating the instance of the service provider framework XFS service provider base class and service provider framework request object further comprises instantiating a specific instance of the service provider's service provider request object derived from an XFS service provider base class service provider request object class hierarchy.

9. The method of claim 7, wherein instantiating the instance of the service provider framework XFS service provider base class and service provider framework request object further comprises instantiating the instance of the service provider framework XFS service provider base class and at least one instance of the service provider framework request object defined for a request class selected from a plurality of request classes derived from one another.

10. The method of claim 7, wherein instantiating the instance of the service provider framework XFS service provider base class and framework request object further comprises instantiating the instance of the service provider framework XFS service provider base class and at least one instance of the framework request object defined for a request class selected from a group of request classes consisting of a SpiRequest class, a SpiAsyncRequest class, a request specific class, and a service provider specific request class.

11. The method of claim 1, wherein obtaining the transaction terminal device service by the service provider further comprises invoking at least one virtual method within a derived object of the service provider by the service provider framework through class inheritance to allow the service provider to perform processing unique to the transaction terminal device service.

12. The method of claim 1, wherein obtaining the transaction terminal device service by the service provider further comprises processing the request in at least one of an immediate processing part and a deferred processing part.

13. The method of claim 12, wherein processing the request in the immediate processing part further comprises performing immediate processing in a thread used by the XFS manager when invoking an entry point of the service provider.

14. The method of claim 12, wherein processing the request in the immediate processing part further comprises performing parameter verification by a service provider framework in an immediate processing method of the service provider framework.

15. The method of claim 12, wherein processing the request in the immediate processing part further comprises invoking a spImmediateProcessing() method within the service provider's derived class by the service provider framework.

16. The method of claim 15, wherein invoking the spImmediateProcessing() method within the service provider's derived class further comprises returning a return code from spImmediateProcessing() to the XFS manager.

17. The method of claim 12, wherein processing the request in the immediate processing part further comprises determining a code by a service provider framework for return to the XFS manager.

18. The method of claim 12, wherein processing the request in the immediate processing part further comprises processing at least one of a WFPCancelAsyncRequest request, a WFPSetTraceLevel request, and a WFPUnload request as a process immediate request by a service provider framework.

19. The method of claim 12, wherein processing the request in the deferred processing part further comprises placing a request object on a deferred processing queue.

20. The method of claim 19, wherein processing the request in the deferred processing part further comprises popping the request object from the deferred processing queue by a deferred processing queue manager thread.

21. The method of claim 20, wherein processing the request in the deferred processing part further comprises processing the request in a deferred processing thread created by the deferred processing queue manager.

22. The method of claim 12, wherein processing the request in the deferred processing part further comprises performing deferred processing for the request that is common to all service providers by a service provider framework.

23. The method of claim 12, wherein processing the request in the deferred processing part further comprises invoking a spDeferredProcessing() method within a derived class of the service provider by a service provider framework.

24. The method of claim 23, wherein processing the request in the deferred processing part further comprises performing all processing necessary to satisfy the request within the spDeferredProcessing method by the service provider.

25. The method of claim 24, wherein processing the request in the deferred processing part further comprises posting a request complete event by the service provider.

26. The method of claim 25, wherein processing the request in the deferred processing part further comprises terminating a thread and deleting a request object.

27. The method of claim 12, wherein processing the request in the deferred processing part further comprises posting a request complete event by a service provider framework.

28. The method of claim 1, wherein obtaining the device service by the service provider further comprises accessing at least one request parameter by the service provider.

29. The method of claim 28, wherein accessing the request parameter by the service provider further comprises accessing the parameter for a request selected

from a group of requests consisting of a WFPCancelAsyncRequest request, a WFPClose request, a WFPDeregister request, a WFPExecute request, a WFPGetInfo request, a WPFLock request, a WFPOpen request, a WFPRegister request, a WFPSetTraceLevel request, a WFPUnloadService request, and a WFPUnlock request.

30. The method of claim 29, wherein accessing the request parameter for the WFPCancelAsyncRequest request by the service provider further comprises accessing at least one of an hService parameter and a reqID parameter.

31. The method of claim 29, wherein accessing the request parameter for the WFPClose request by the service provider further comprises accessing at least one of an hService parameter, an hWnd parameter, and a reqID parameter.

32. The method of claim 29, wherein accessing the request parameter for the WFPDeregister request by the service provider further comprises accessing at least one of an hService parameter, a dwEventClass parameter, an hWndReg parameter, an hWnd parameter, and a reqID parameter.

33. The method of claim 29, wherein accessing the request parameter for the WFPExecute request by the service provider further comprises accessing at least one of an hService parameter, a dwCommandData parameter, an IpCommandData parameter, a dwTimeOut parameter, an hWnd parameter, and a reqID parameter.

34. The method of claim 29, wherein accessing the request parameter for the WFPGetInfo request by the service provider further comprises accessing at least one of an hService parameter, a dwCagegory parameter, an IPQueryDetails parameter, a dwTimeOut parameter, an hWnd parameter, and a reqID parameter.

35. The method of claim 29, wherein accessing the request parameter for the WFPOpen request by the service provider further comprises accessing at least one of an hService parameter , an IpszLogicalName parameter , an hApp parameter , an IpszApplicationID parameter, a dwTraceLevel parameter , a dwTimeOut parameter , an hWnd parameter , a reqID parameter , an hPprovider parameter , a

dwSPVersionsRequired parameter , an IpSPIVersion parameter , a dwSrvcVersionsRequired parameter , and an IpSrvcVersion parameter .

36. The method of claim 29, wherein accessing the request parameter for the WFPRegister request by the service provider further comprises accessing at least one of an hService parameter, a dwEventClass parameter, an hWndReg parameter, an hWnd parameter, and a reqID parameter.

37. The method of claim 29, wherein accessing the request parameter for the WFPSetTraceLevel request by the service provider further comprises accessing at least one of an hService parameter and a dwTraceLevel parameter.

38. A system for obtaining at least one device service on a self-service transaction terminal using a service provider framework, comprising:

means for receiving data by a self-service transaction terminal application indicative of a need for obtaining a transaction terminal device service;

means for issuing a request by the transaction terminal application to an XFS manager to get the transaction terminal device service;

means for translating the request by the XFS manager for processing by a service provider; and

means for obtaining the transaction terminal device service by the service provider.

39. The system of claim 38, wherein the means for receiving the data by the transaction terminal application further comprises means for receiving the data indicative of the need for obtaining the transaction terminal device service in connection with a device selected from a group of transaction terminal devices consisting of a depository, a printer, a card reader, a safe door, a cash dispenser, and a touchscreen.

40. The system of claim 38, wherein the means for issuing the request to the XFS manager further comprises means for making a sub-routine call by the transaction terminal application to the XFS manager to get the transaction terminal device service from a service provider.

41. The system of claim 38, wherein the means for translating the request for processing by the service provider further comprises means for translating a sub-routine call by the XFS manager as an entry point into the service provider for processing by the service provider.

42. The system of claim 41, wherein the means for translating the sub-routine call as an entry point into the service provider further comprises means for instantiating a request object associated with the request.

43. The system of claim 38, wherein the means for translating the request for processing by the service provider further comprises means for implementing the service provider.

44. The system of claim 43, wherein the means for implementing the service provider further comprises means for instantiating an instance of a service provider framework XFS service provider base class and at least one instance of a service provider framework request object required to process the request.

45. The system of claim 44, wherein the means for instantiating the instance of the service provider framework XFS service provider base class and service provider framework request object further comprises means for instantiating a specific instance of the service provider's service provider request object derived from an XFS service provider base class service provider request object class hierarchy.

46. The system of claim 44, wherein the means for instantiating the instance of the service provider framework XFS service provider base class and service provider framework request object further comprises means for instantiating the instance of the service provider framework XFS service provider base class and at least one instance of the service provider framework request object defined for a request class selected from a plurality of request classes derived from one another.

47. The system of claim 44, wherein the means for instantiating the instance of the service provider framework XFS service provider base class and

framework request object further comprises means for instantiating the instance of the service provider framework XFS service provider base class and at least one instance of the framework request object defined for a request class selected from a group of request classes consisting of a SpiRequest class, a SpiAsyncRequest class, a request specific class, and a service provider specific request class.

48. The system of claim 38, wherein the means for obtaining the transaction terminal device service by the service provider further comprises means for invoking virtual methods within derived objects of the service provider by the service provider framework through class inheritance to allow the service provider to perform processing unique to the transaction terminal device service.

49. The system of claim 38, wherein the means for obtaining the transaction terminal device service by the service provider further comprises means for processing the request in at least one of an immediate processing part and a deferred processing part.

50. The system of claim 49, wherein the means for processing the request in the immediate processing part further comprises means for performing immediate processing in a thread used by the XFS manager when invoking an entry point of the service provider.

51. The system of claim 49, wherein the means for processing the request in the immediate processing part further comprises means for performing parameter verification by a service provider framework.

52. The system of claim 49, wherein the means for processing the request in the immediate processing part further comprises means for invoking a spImmediateProcessing() method within the service provider's derived class by the service provider framework.

53. The system of claim 52, wherein the means for invoking the spImmediateProcessing() method within the service provider's derived class

further comprises means for returning a return code from spImmediateProcessing() to the XFS manager.

54. The system of claim 49, wherein the means for processing the request in the immediate processing part further comprises means for determining a code by a service provider framework for return to the XFS manager.

55. The system of claim 49, wherein the means for processing the request in the immediate processing part further comprises means for processing at least one of a WFPCancelAsyncRequest request, a WFPSetTraceLevel request, and a WFPUnload request as a process immediate request by a service provider framework.

56. The system of claim 49, wherein the means for processing the request in the deferred processing part further comprises means for placing a request object on a deferred processing queue.

57. The system of claim 49, wherein the means for processing the request in the deferred processing part further comprises means for popping a request object from a deferred processing queue by a deferred processing queue manager thread.

58. The system of claim 57, wherein the means for processing the request in the deferred processing part further comprises means for processing the request in a deferred processing thread created by the deferred processing queue manager.

59. The system of claim 49, wherein the means for processing the request in the deferred processing part further comprises means for performing deferred processing for the request that is common to all service providers by a service provider framework.

60. The system of claim 49, wherein the means for processing the request in the deferred processing part further comprises means for invoking a spDeferredProcessing() method within a derived class of the service provider by the service provider framework.

61. The system of claim 60, wherein the means for processing the request in the deferred processing part further comprises means for performing all

processing necessary to satisfy the request within the spDeferredProcessing method by the service provider.

62. The system of claim 61, wherein the means for processing the request in the deferred processing part further comprises means for posting a request complete event by the service provider.

63. The system of claim 62, wherein the means for processing the request in the deferred processing part further comprises means for terminating a thread and deleting a request object.

64. The system of claim 49, wherein the means for processing the request in the deferred processing part further comprises means for posting a request complete event by a service provider framework.

65. The system of claim 38, wherein the means for obtaining the device service by the service provider further comprises means for accessing at least one request parameter by the service provider.

66. The system of claim 65, wherein the means for accessing the request parameter by the service provider further comprises means for accessing the parameter for a request selected from a group of requests consisting of a WFPCancelAsyncRequest request, a WFPClose request, a WFPDeregister request, a WFPExecute request, a WFPGetInfo request, a WPFLock request, a WFPOpen request, a WFPRegister request, a WFPSetTraceLevel request, a WFPUnloadService request, and a WFPUnlock request.

67. The system of claim 66, wherein the means for accessing the request parameter for the WFPCancelAsyncRequest request by the service provider further comprises means for accessing at least one of an hService parameter and a reqID parameter.

68. The system of claim 66, wherein the means for accessing the request parameter for the WFPClose request by the service provider further comprises means for accessing at least one of an hService parameter, an hWnd parameter, and a reqID parameter.

69. The system of claim 66, wherein the means for accessing the request parameter for the WFPDeregister request by the service provider further comprises means for accessing at least one of an hService parameter, a dwEventClass parameter, an hWndReg parameter, an hWnd parameter, and a reqID parameter.

70. The system of claim 66, wherein the means for accessing the request parameter for the WFPExecute request by the service provider further comprises means for accessing at least one of an hService parameter, a dwCommandData parameter, an IpCommandData parameter, a dwTimeOut parameter, an hWnd parameter, and a reqID parameter.

71. The system of claim 66, wherein the means for accessing the request parameter for the WFPGetInfo request by the service provider further comprises means for accessing at least one of an hService parameter, a dwCagegory parameter, an IPQueryDetails parameter, a dwTimeOut parameter, an hWnd parameter, and a reqID parameter.

72. The system of claim 66, wherein the means for accessing the request parameter for the WFPOpen request by the service provider further comprises means for accessing at least one of an hService parameter , an IpszLogicalName parameter , an hApp parameter , an IpszApplicationID parameter, a dwTraceLevel parameter , a dwTimeOut parameter , an hWnd parameter , a reqID parameter , an hPprovider parameter , a dwSPVersionsRequired parameter , an IpSPIVersion parameter , a dwSrvcVersionsRequired parameter , and an IpSrvcVersion parameter .

73. The system of claim 66, wherein the means for accessing the request parameter for the WFPRegister request by the service provider further comprises means for accessing at least one of an hService parameter, a dwEventClass parameter, an hWndReg parameter, an hWnd parameter, and a reqID parameter.

74. The system of claim 66, wherein the means for accessing the request parameter for the WFPSetTraceLevel request by the service provider further

comprises means for accessing at least one of an hService parameter and a dwTraceLevel parameter.

75. A method for obtaining at least one device service on a self-service transaction terminal using a service provider framework, comprising:

receiving data by the self-service transaction terminal application indicative of a need for the performance of a transaction terminal device function selected from a group of device functions consisting of a depository function, a printer function, a card reader function, a safe door function, a cash dispenser function, and a touchscreen function;

making a sub-routine call by the transaction terminal application to a lower level layer of central transaction terminal monitoring and management application software to request device service from a service provider;

translating the sub-routine call at the lower level layer into a function category request by an XFS manager as an entry point into the service provider for processing by the service provider, the function request being selected from a group of function requests consisting of a WFPCancelAsyncRequest request, a WFPClose request, a WFPDeregister request, a WFPExecute request, a WFPGetInfo request, a WPFLock request, a WFPOpen request, a WFPRegister request, a WFPSetTraceLevel request, a WFPUnloadService request, and a WFPUnlock request; and

returning a result of the function category request to the application by the service provider.